

CLIPPEDIMAGE= JP402004288A

PAT-NO: JP402004288A

DOCUMENT-IDENTIFIER: JP 02004288 A

TITLE: IMAGE PROCESSOR AND METHOD FOR MANAGING ITS FONT FILE

PUBN-DATE: January 9, 1990

INVENTOR-INFORMATION:

NAME

YAMASHITA, TAKETOSHI

ASSIGNEE-INFORMATION:

NAME

COUNTRY

FUJI XEROX CO LTD

N/A

APPL-NO: JP63153890

APPL-DATE: June 22, 1988

INT-CL (IPC): G09G001/02;G06F015/64 ;G09G005/36 ;H04N001/387

US-CL-CURRENT: 345/471

ABSTRACT:

PURPOSE: To easily check errors by providing a font file with at least one check sum, sum-checking font data based on the check sum and detecting the presence or absence of the data errors of the font file.

CONSTITUTION: The font file is provided with the check sum and sum-checking is performed based on the check sum so as to detect the presence or absence of the data errors of the font file. Thus, even when the font file is destroyed by operational errors, etc., it is easy to know places where the data errors occur. Therefore, error-checking can easily be performed. When the font file is loaded from memory devices, such as a hard disk, onto an image memory, sum-checking is performed. Thus, the data errors of the font file can be detected when filing or loading is performed.

COPYRIGHT: (C)1990,JPO&Japio

## ⑫ 公開特許公報(A) 平2-4288

⑤ Int. Cl.<sup>5</sup>

識別記号

庁内整理番号

④ 公開 平成2年(1990)1月9日

G 09 G 1/02  
G 06 F 15/64  
G 09 G 5/36  
H 04 N 1/387

L 8121-5C  
F 8419-5B  
8839-5C  
8839-5C

審査請求 未請求 請求項の数 2 (全6頁)

⑥ 発明の名称 イメージ処理装置及びそのフォントファイル管理方法

⑦ 特 願 昭63-153890

⑦ 出 願 昭63(1988)6月22日

⑦ 発 明 者 山 下 武 利 埼玉県岩槻市大字岩槻1275番地 富士ゼロックス株式会社  
岩槻事業所内⑦ 出 願 人 富士ゼロックス株式会 東京都港区赤坂3丁目3番5号  
社

⑦ 代 理 人 弁理士 木村 高久

## 明 細 書

## 1. 発明の名称

イメージ処理装置及びそのフォントファイル  
管理方法

## 2. 特許請求の範囲

(1) イメージデータとフォントファイルを同  
一のイメージメモリ内に展開して処理するイメー  
ジ処理装置において、少なくとも一つのチェックサムを設けたフォ  
ントファイルと、前記チェックサムに基づいてフォントデータの  
サムチェックを行ない、フォントファイルのデー  
タエラーの有無を検出する検出手段と、

を具えたことを特徴とするイメージ処理装置。

(2) イメージデータとフォントファイルを同  
一のイメージメモリ内に展開して処理するイメー  
ジ処理装置において、前記イメージメモリ内でイメージデータとフォ  
ントデータの合成及び加工を行なう工程と、前記フォントファイルに設けられたチェックサ  
ムに基づいて、フォントデータのサムチェックを  
実行する工程と、を具え、前記サムチェックによりエラーが検出さ  
れたときは、イメージメモリから画像出力部への  
イメージデータの送出を中断するようにしたこと  
を特徴とするイメージ処理装置のフォントファイ  
ル管理方法。

## 3. 発明の詳細な説明

(産業上の利用分野)

この発明は、入力したイメージデータと漢字  
等のフォントデータを合成・加工して出力するイ  
メージ処理装置と、そのフォントファイル管理方  
法に関する。

(従来の技術)

第4図は、従来のこの種のイメージ処理装置  
の構成ブロック図である。第4図において、CP  
U(中央処理装置)3は、画像入力部1から入力  
したイメージデータを、イメージメモリ5のイメ  
ージ領域に蓄積すると共に、ハードディスク4か

ら読み取ったフォントファイルを、イメージメモリ5のフォント領域に蓄積する。そして、イメージメモリ5に蓄積されたイメージとフォントをメモリ内で合成・加工し、画像出力部2において出力させる。

(発明が解決しようとする課題)

ところで、上記イメージ処理装置では、イメージメモリ内にイメージとフォントファイルをデータごとに格納しているが、イメージ領域とフォント領域に境界を設けてしまうと、片方の領域がオーバーフローした時に他の領域を使用できなくなるので、各領域を仕切るための境界は設けられていない。このため、例えば、イメージメモリからフォントファイルをコピーして取り出す時に、領域の指定ミスによりフォントファイルの一部を破壊してしまうことがあった。この場合、破壊された部分はデータエラーとなり、CRTディスプレイ等に表示すると、画像乱れとなって表われる。しかし、従来は、データのエラー箇所を確認することができなかったため、エラー箇所を見つける

ために、フォントファイルのデータを最初から見直さなければならず、エラーチェックが煩雑になるという問題点があった。

この発明は、上記問題点に鑑みなされたもので、フォントファイルのデータエラー箇所を検出し、エラーチェックを容易に行なうことができるイメージ処理装置及びそのフォントファイル管理方法を提供することを目的とする。

(課題を解決するための手段)

上記目的を達成するため、この発明に係るイメージ処理装置は、フォントファイルに少なくとも一つのチェックサムを設けると共に、このチェックサムに基づいてフォントデータのサムチェックを行ない、フォントファイルのデータエラーの有無を検出する検出手段を設けたことを特徴とする。

また、上記イメージ処理装置のフォントファイル管理方法は、イメージメモリ内でイメージデータとフォントデータの合成及び加工を行なう工程と、フォントファイルに設けられたチェックサム

に基づいて、フォントデータのサムチェックを実行する工程とを具え、前記サムチェックによりエラーが検出されたときは、イメージメモリから画像出力部へのイメージデータの送出を中断するようにしたことを特徴とする。

(作用)

イメージメモリ内においてイメージとフォントの合成・加工を行なった後、フォントデータに設けたチェックサムを用いて、フォントデータのサムチェックを行なう。そして、サムチェック終了後、フォントデータの合計があらかじめ設定されたサム値に対して正常か否かを検出手段により判断する。ここでデータエラーが発生していなければ画像出力部へイメージデータを送出し、データエラーが検出されたときは、そのデータエリアに関するデータの画像出力部への送出を中断する。したがって、データエラーの発生した箇所を速やかに検出することができる。

(実施例)

以下、この発明の実施例を図面と共に詳細に

説明する。

第2図は、この発明に係るイメージ処理装置の一実施例を示す構成ブロック図である。図において、11はイメージメモリであり、イメージデータとフォントファイルが格納される。ここで用いられるフォントファイルの一例を第1図に示す。フォントファイルは、4096ワードを一単位とする複数のバンクから構成され、0バンク目にチェックサムテーブル、1～3バンク目にフォントコードを示すフォントコードテーブル、4～11バンク目にフォントデータが配置されている。前記0バンク目のチェックサムテーブルのアドレス(1～11)は、フォントファイルの各バンク番号1～11に対応している。例えば、バンク3のチェックサム(SUM-3)は、チェックサムテーブルのアドレス3に入っている。なお、バンク0に対応するアドレス0には、チェックサムは入っていない。また、チェックサムテーブルのアドレス12～4095にはデータが入っていないものとする。また、前記イメージメモリ11は、

上位12ビットをバンク番号(0~11)、下位12ビットをバンク内アドレス(0~4095)として、4096×4096(ワード)の空間に分割されている。なお、1ワードは16ビットで構成されている。12はフォントファイルが記憶されたハードディスクであり、ハードディスクコントローラ15によりデータの入・出力が制御される。13は合成・加工されたイメージデータを出力するためのプリンタであり、その出力はプリンタコントローラ16により制御される。なお、この例では画像出力部にプリンタを用いたが、出力装置としてはCRTディスプレイやXYプロッタ等を用いてもよい。14はイメージデータを入力するためのイメージリーダーであり、データの入力にはイメージリーダーコントローラ17により制御される。

18はROMであり、後述するサムチェックの処理手順のフローチャートに基づいたプログラム等が記憶されている。19はRAMであり、サムチェックや出力を行なう前のイメージデータが格

納される。20は各部の制御を行なうCPUであり、イメージ処理においてはイメージメモリ11でのイメージとフォントの合成・加工を制御する。また、前記ROM18に記憶された処理プログラムに基づいて、フォントデータのサムチェックを実行する。CPU20でのサムチェックは、次式(1)の計算結果に基づいて行なわれる。

判定値 =

$$\text{adr}(0) + \text{adr}(1) + \text{adr}(2) + \dots + \text{adr}(4095) + (\text{SUM} - j) \quad \dots (1)$$

この例では、算出された判定値が0のときにデータエラーなしと判断する。すなわち、jバンク目の全データの合計と、そのバンクのサム値SUM-jを加算して0であればフォントファイルが正常であると判断する。

21はアドレスコントローラ(シーケンサ)であり、前記プリンタコントローラ16、イメージリーダーコントローラ17等の入出力制御装置や、イメージメモリ11とCPU20との間のデータ転送に使用される。22はデータレジスタであり、

CPUバスとイメージバスとの間でRAM19やイメージメモリ11に格納されたバンクのアドレス番号を順次指定する。

また、23はホストコンピュータ、24は前記ホストコンピュータ23とCPUバスを結ぶホストI/F(インターフェース)である。

次に、上記イメージ処理装置におけるサムチェックの処理手順を第3図のフローチャートに基づいて説明する。

まずサムチェックを行なう前に、CPU20は、イメージリーダー14から入力されたイメージデータをイメージメモリ11のイメージ領域へ番換させると共に、ハードディスク12に記憶されたフォントファイルを読み出し、イメージメモリ11のフォント領域へ番換させる。そして、このイメージメモリ11内でイメージとフォントの合成・加工を行なう。

次に、CPU20は、イメージメモリ11のバンクjに初期値1を与える(ステップ101)。これによりバンクjが指定される。続いて、イメ

ージメモリ11の0バンク目のアドレスjに格納されたSUM-j(チェックサム)を取り出し、SUMとする(ステップ102)。そして、イメージメモリ11のバンクjのアドレスiを0にクリアする(ステップ103)。次に、前記SUM(SUM-j)とイメージメモリ11のバンクjのアドレスi(ここではアドレス0)のデータを加算し、その値をさらにSUMに加える(ステップ104)。次に、バンクjのアドレスiに1を加え、これを新たなアドレスiとしてSUM+IM(j, i)を行なう(ステップ105)。すなわち、次のアドレス1のデータがSUMに加えられることになる。そして、上記処理を順次繰り返し、i=4096となったときに(ステップ106)、SUM←SUM+IM(j, 0)+IM(j, 1)+...+IM(j, 4095)におけるSUMが0となるかどうかを判断する(ステップ107)。ここで、SUM=0であれば、そのバンクのデータを正常と判断し、バンクjに1を加え、新たなバンクjについて同様にSUM+

IM(j, i)を行なう(ステップ108)。そして、上記処理を各バンクについて順次繰り返し、バンクjmax(この例では11)となったかどうかを判断し(ステップ109)、jmaxであればエラーなしと判定して処理を終了する。また、jmaxでなければjmaxとなるまで処理を繰り返す。

一方、ステップ107においてSUM=0とならない場合は、バンクjのサムチェックエラーと判定し(ステップ110)、エラー処理として、外部にエラー表示を行なうと共に、イメージメモリ11からプリンタ13へのイメージデータの送出を中断する。そして、装置をストップさせ、イメージメモリ内のイメージデータをクリアし、フォントファイルの再ロードを行なう(ステップ111)。

このように、各バンクごとにアドレス0~4095までの全てのデータを順次加算し、これをさらにチェックサムと合計してサムチェックを行なうことにより、各バンクごとにデータエラーの有

無を検出することができる。したがって、使用者はデータエラーが検出されたときは、該当するバンクのデータだけを見直せばよいので、エラーチェックを容易に行なうことができる。

なお、上記実施例では、チェックサムを論理和=0で説明したが、排他的論理和(EXOR)、CRC等の方法を用いてもよい。

また、この実施例では、フォントファイルの先頭(0バンク目)に全バンク分のチェックサムテーブルが設けられているが、各バンクの先頭もしくは終りにそのバンク分のチェックサムを設けてもよい。

さらに、この実施例では、フォントファイルを4096ワード単位のバンクに分割しているが、このワード数は増減させてもよいし、フォントファイルが複数ある場合は1つのフォントファイルに1つのチェックサムを設定し、ファイルごとにデータエラーの有無をチェックするようにしてもよい。

(発明の効果)

以上説明したように、この発明によれば、フォントファイルにチェックサムを設け、このチェックサムに基づいてサムチェックを行ない、フォントファイルのデータエラーの有無を検出するようにしたため、操作ミス等によりフォントファイルを破壊してしまっても、データエラーの発生した箇所を簡単に知ることができるので、エラーチェックを容易に行なうことが可能となる。

また、フォントファイルをハードディスク等の記憶装置からイメージメモリにロードする際に、このサムチェックを行なうことにより、ファイル時又はロード時のフォントファイルのデータエラーを検出することができる。

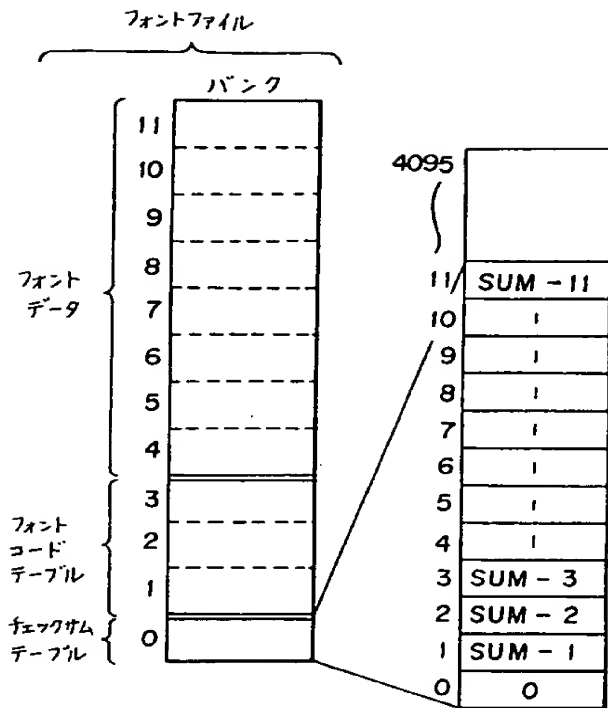
#### 4. 図面の簡単な説明

第1図はこの発明に係るイメージ処理装置のフォントファイルの一実施例を示す概念図、第2図はこの発明に係るイメージ処理装置の一実施例を示す構成ブロック図、第3図はサムチェックの処理手順を示すフローチャート、第4図は従来のイメージ処理装置の構成ブロック図である。

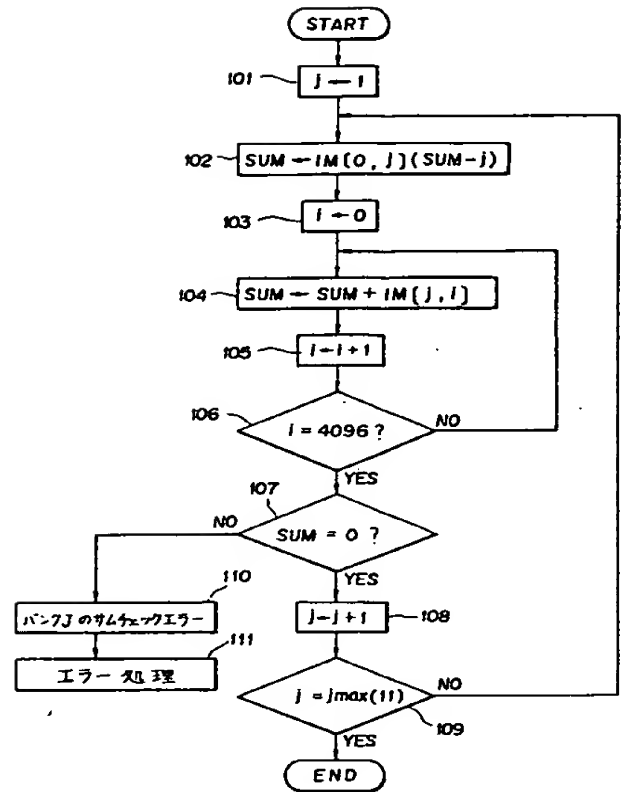
11…イメージメモリ、12…ハードディスク、13…プリンタ、14…イメージリーダー、18…ROM、19…RAM、20…CPU(中央処理装置)。

出願人代理人 木村 昌久

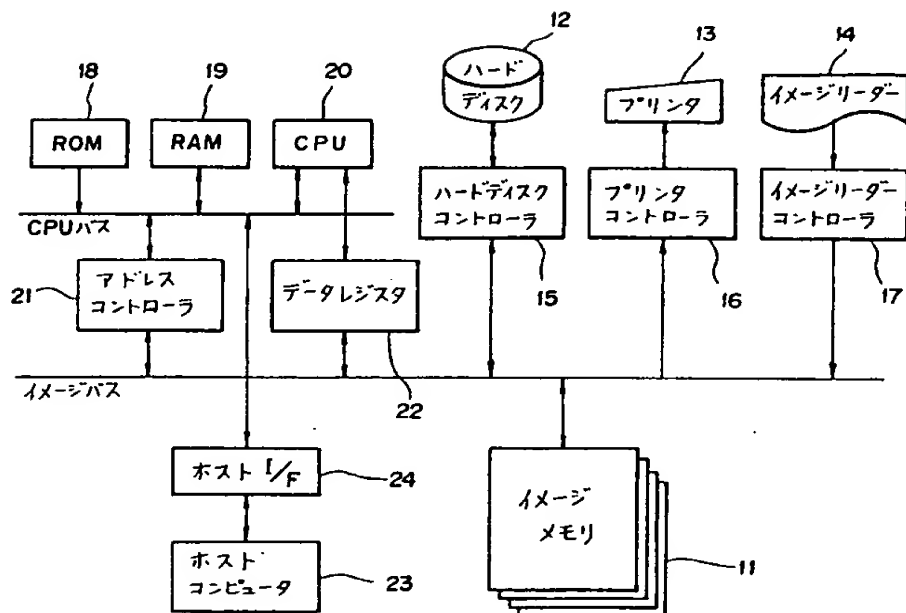




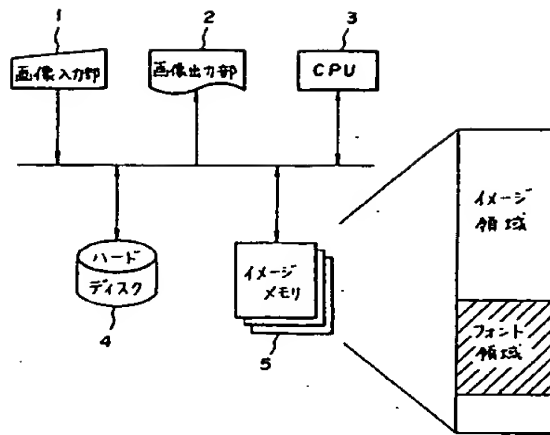
第 1 図



第 3 図



第 2 図



第 4 図